

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer implemented method comprising:

determining whether an intermediate driver is present in memory, wherein determining whether an intermediate driver is present in memory comprises:

receiving a message from a calling process, wherein receiving a message from a calling process comprises receiving an event control block from said calling process; and

determining, on the basis of said message, whether said calling process is an intermediate driver; and

adaptively controlling a persistency of a device driver present in memory on the basis of whether said intermediate driver is present in memory.

- 2-3. (Canceled)

4. (Original) The method of claim 1 further comprising setting said device driver to be non-persistent when an intermediate driver is not present in memory.

5. (Original) The method of claim 1 further comprising setting said device driver to be persistent when an intermediate driver is present in memory.

6. (Original) The method of claim 1 further comprising:  
  
determining whether persistency for said device driver is pre-specified; and  
  
if said persistency is pre-specified, setting said device driver to operate consistently with said pre-specified persistency.

7. (Original) The method of claim 1 further comprising:  
  
specifying a default persistency for said device driver; and  
  
setting said device driver to said default persistency on the basis of whether an intermediate driver is present in memory.

8. (Original) The method of claim 1 further comprising:  
  
specifying a default persistency for said device driver;  
  
if an intermediate driver is present in memory, determining whether persistency of said driver is under automatic control; and  
  
if persistency of said device driver is under automatic control setting said driver to be persistent.

9. (Currently Amended) A server comprising:

a network-interface card;

a memory;

a device driver for interfacing with said network-interface card, said device driver being configured to:

adaptively set its persistency in said memory on the basis of whether an intermediate driver is present in said memory,

determine whether persistency for said device driver is pre-specified, and

if said persistency is pre-specified, set said device driver to operate consistently with said pre-specified persistency.

10. (Original) The server of claim 9 wherein said device driver is configured to be persistent in said memory when an intermediate driver is present in said memory.

11. (Original) The server of claim 9 wherein said device driver is configured to be non-persistent in said memory when an intermediate driver is not present in said memory.

12. (Original) The server of claim 9 wherein said device driver is configured to receive a message from a calling process in said memory and to determine, on the basis of said message, whether said calling process is an intermediate driver.

13. (Original) The server of claim 9 further comprising configuration data for said device driver.

14. (Original) The server of claim 13 wherein said configuration data comprises data indicative of a pre-specified persistency state for said device driver.

15. (Original) The server of claim 13 wherein said configuration data comprises instructions to disable automatic persistency determination by said device driver.

16. (Currently Amended) A network interface for enabling a computer to send and receive messages over a network, said network interface comprising:

a network-interface card in communication with a memory element of said computer;

a device driver for controlling said network-interface card in response to messages received from a calling process, said device driver being configured to set its persistency in said memory on the basis of said calling process wherein said device driver is configured to:

determine whether persistency for said device driver is pre-specified; and

if said persistency is pre-specified, set said device driver to operate consistently with said pre-specified persistency.

17. (Original) The network interface of claim 16 wherein said device driver is configured to determine, on the basis of a message received from said calling process, whether said calling process is an intermediate driver.

**18.** (Original) The network interface of claim **16** wherein said device driver is configured to set said device driver to be non-persistent when said calling process is not an intermediate driver.

**19.** (Original) The network interface of claim **16** wherein said device driver is configured to set said device driver to be persistent when said calling process is an intermediate driver.

**20.** (Canceled)

**21.** (Original) The network interface of claim **16** wherein said device driver is set to a default persistency when said calling process is not an intermediate driver.

**22.** (Original) The network interface of claim **16** wherein said automatic control of persistency status by a device driver can be disabled and disabled and said device driver is configured to be persistent when said automatic control of persistency status is enabled and said calling process is an intermediate driver.

**23.** (Currently Amended) A machine-readable medium having encoded thereon software for controlling persistency of a device driver in memory, said software comprising instructions for:

determining whether an intermediate driver is present in memory; and

adaptively setting said persistency of said device driver on the basis of whether said intermediate driver is present in memory;

determining whether persistency for said device driver is pre-specified; and

if said persistency is pre-specified, setting said device driver to operate consistently with said pre-specified persistency.

**24.** (Original) The machine-readable medium of claim **23** wherein said instructions for determining whether an intermediate driver is present in memory comprise instructions for:

receiving a message from a calling process; and

determining, on the basis of said message, whether said calling process is an intermediate driver.

**25.** (Original) The machine-readable medium of claim **23** wherein said instructions for receiving a message from a calling process comprise receiving an event control block from said calling process.

**26.** (Original) The machine-readable medium of claim **23** wherein said software further comprises instructions for setting said device driver to be non-persistent when an intermediate driver is not present in memory.

**27.** (Original) The machine-readable medium of claim **23** wherein said software further comprises instructions for setting said device driver to be persistent when an intermediate driver is present in memory.

**28.** (Canceled)

**29.** (Original) The machine-readable medium of claim **23** wherein said software further comprises instructions for:

specifying a default persistency for said device driver;

if an intermediate driver is present in memory, determining whether persistency of said driver is under automatic control; and

if persistency of said device driver is under automatic control setting said driver to be persistent.